IBM Österreich

BusinessContact/TCP

Win32-Client Version 2.21

Benutzerhandbuch

http://www.ibm.com/at/services/buscon

Dieses Dokument wurde erstellt von:
Mag. Michael Höller
Mail: m.hoeller@at.ibm.com
Erstellt am 30. Juni 1998
Letzte Änderung am 27. November 2000

Inhaltsverzeichnis

1. Einleitung 5

1.1. Allgemeines 5

1.2. Erweiterungen der vorliegenden Version 5

1.2.1. Version 1.1 5

1.2.2. Version 1.2 6

1.2.3. Version 1.32 6

1.2.4. Version 2.00 6

1.2.5. Version 2.01 7

1.2.6. Version 2.20 7

1.3. Systemumgebung 7

2. Installation 9

2.1. Wählzugang für Windows installieren 9

2.1.1. Hinweise 9

2.1.2. Installationsanweisungen für Windows 95 10

2.1.3. Modem installieren 10

2.1.4. DFÜ-Netzwerk installieren 10

2.1.5. DFÜ-Adapter und TCP/IP-Protokoll installieren 10

2.2. Konfiguration der Wählzugangssoftware 11

2.2.1. DFÜ-Netzwerk konfigurieren 11

2.2.2. TCP/IP-Protokoll konfigurieren 12

2.2.3. Besonderheiten bei der Installation unter Windows 98 12

2.2.4. Besonderheiten bei der Installation unter Windows NT 12

2.3. Prüfung der Verbindung 13

2.4. Installation von BC/TCP 14

2.5. Upgrade von vorhergehenden Versionen 14

3. BC/TCP-Bedienung 15

3.1. User Interface 15

3.2. Kommunikationsparameter 17

3.2.1. Network-Properties 17

3.2.2. Mailbox-Properties 18

3.2.3. Send-Properties (für Cmd. Send) 19

3.2.4. Receive-Properties (für Cmd. Receive, Reactivate) 20

3.2.5. Protocol-Properties (für alle Cmd.) 21

3.2.6. EDIPAC-Properties 22

3.2.7. Properties for creating an EDIFACT-Envelope 23

3.2.8. Properties for deleting the EDIFACT-Envelope 23

3.2.9. Selection-Properties 24

3.2.10. InformationExchange-Sender 25

3.2.11. InformationExchange-Receiver 25

3.2.12. Other Properties 26

3.2.13. Zip-Properties 27

3.2.14. Protocollog-Properties 28

3.2.15. About 28

3.3. Test von BC/TCP 29

4. Automatische Steuerung 30

4.1. Initialisierungsdatei....30
4.2. Befehlsdatei....30
4.3. Programmparameter....32
4.4. Einbindung von BC/TCP in andere Programme....32
4.5. Aufbau der Befehlsdatei....33
4.6. Befehle....34
4.7. Parameter....34
4.7.1. Transferparameter....34
4.7.2. Verbindungsparameter....35
4.7.3. Loginparameter....35
4.7.4. Sendeparameter....36
4.7.5. Empfangsparameter....36
4.7.6. Parameter zum Empfangen des Protokolls....37
4.7.7. EDIPAC-Parameter....37
4.7.8. Parameter für die automatische Kuvertierung....37
4.7.9. Parameter für die automatische Dekuvertierung....39
4.7.10. Parameter für die Datenselektion....39
4.7.11. Parameter für InformationExchange (Sender)....40
4.7.12. Parameter für InformationExchange (Receiver)....40
4.7.13. Spezialparameter....41
4.7.14. Parameter für Datenkompression....42
4.7.15. Parameter für den Protokolllog....42
4.7.16. Lizenzparameter....42
4.7.17. Programmparameter....42
5. Protokoll des BC/TCP-Gateways....44
5.1. Protokoll für Löschen und Abfragen....44
5.2. Protokoll für Senden....45
5.3. Protokoll für Holen und Reaktivieren....46
5.4. Erläuterungen zum Protokoll....46
5.5. Client-Status....47

1. Einleitung

1.1. Allgemeines

BC/TCP ist die Abkürzung für **IBM BusinessContact TCP/IP**, ein Datenaustauschprogramm für das IBM-Mailboxsystem **BusinessContact** (bzw. **ECODEX** für EAN-Kunden).

Die Internet-Website von **IBM BusinessContact** finden Sie unter http://www.ibm.com/at/services/buscon . Dort finden Sie die **Links Software -> BC/TCP**, wo Informationen über die aktuelle Version von **BC/TCP** zur Verfügung stehen und Sie sich diese auch herunterladen können.

Wenn Sie in die **BC/TCP-Mailinglist** aufgenommen werden wollen, so senden Sie bitte eine entsprechende e-mail an den Autor dieses Handbuchs. Sie erhalten dann laufend Informationen über Updates von **BC/TCP**.

Das vorliegende Handbuch ist für die **Version 2.20** des Win32-Clients gültig. Es handelt sich dabei um einen 32-Bit-Client für Microsoft Windows 95/98 und Windows NT.

1.2. Erweiterungen der vorliegenden Version

In diesem Kapitel werden die Änderungen und Zusätze angeführt, die in jeder neuen Version des BC/TCP-Clientprogrammes implementiert wurden.

1.2.1. Version 1.1

- Das ausführbare Programm wurde in eine DLL-umgewandelt, um eine leichtere Einbindung des BC/TCP-Clients in andere Programme zu ermöglichen.
- Das Entzippen wird über einen DLL-Aufruf realisiert.
- Die Ergebnisdatei einer Commandfile-Abarbeitung wird im Programmverzeichnis gespeichert.
- Tests unter Windows 98 wurden durchgeführt.

1.2.2. Version 1.2

- Windowverhalten der DLL verändert, damit ist jetzt das BC/TCP-Icon in der Windows-Taskbar ersichtlich.
- Automatisches Uppercase für bestimmte Felder (z.B. Username).
- Rename-Warning auch bei Protokoll eingebaut.
- Timeout-Fehler bei Rename-Warning ausgebessert.
- Versionsnummer auch in der Titelleiste ersichtlich.
- Quickhints für alle Felder hinzugefügt.

1.2.3. Version 1.32

- Kuvertiersoftware EDIPAC wird mitgeliefert
- Bei mehreren Kommandos in der Commanddatei wird nur einmal gewählt. Wichtig dabei ist, dass zwischen den einzelnen Befehlen während einer Commandfile-Abarbeitung kein Fileviewer-Fenster offen ist, weil dies das Programm in einen Pause-Status versetzen würde und somit unnötige Telefonkosten für Sie anfallen sollten.
- Fileviewer überarbeitet (Print- und Fontfunktionen eingebaut, Dateigrößenlimit ausgeschaltet)
- Parameter "Check SEDAS" dahingehend erweitert, dass bei "Holen EDIFACT" auch einzelne Sendungen empfangen werden können.
- Bugfix: UserMessageClass in INI- oder Commanddatei wird richtig übernommen
- IE-Panels überarbeitet
- Mauszeigerverhalten überarbeitet.

1.2.4. Version 2.00

- Der Aufruf der Kuvertierfunktionen (kuvertieren/dekuvertieren) wurde direkt in das Clientprogramm eingebaut (3 neue Panels). Die Kuvertierfunktionen werden von einem eigenständigen Programm (EDIPAC oder BCPAC) durchgeführt. Beim Senden kann mit dieser Funktion ein EDIFACT-Kuvert automatisch erstellt werden, beim Empfangen und Reaktivieren kann das empfangene EDIFACT-Kuvert automatisch gelöscht werden.

- Änderung der Panel-Bezeichnungen "Connection" zu "Network" und "BC-Login" zu "Mailbox".
- Timing beim Versenden des Commit-Befehles verbessert.

1.2.5. Version 2.01

- Kommunikationshandling verbessert. Dadurch werden etwaige Error #47 und Error #77 verhindert.
- Zip-Handling verbessert.
- Logeintragungen verkleinert (Prozentangaben).

1.2.6. Version 2.20, 2.21

- Kompiliert unter VisualStudio ServicePack 4
- Ressourcenhandling verbessert. Dadurch wird der Prozessor bei Nicht-Aktivität des Programmes nicht mehr belastet.
- Neuer Parameter „Extra Parameter“ (EXTRAPAR). Dadurch wird BC/TCP auch für zukünftige Service-Erweiterungen verwendbar.
- Änderung im Handbuch: Systemumgebung (nicht mehr RS/6000 sondern OS/390 USS)

1.3. Systemumgebung

Für den Benutzer ergibt sich durch den Einsatz dieses Windows-Clients folgende Systemumgebung:

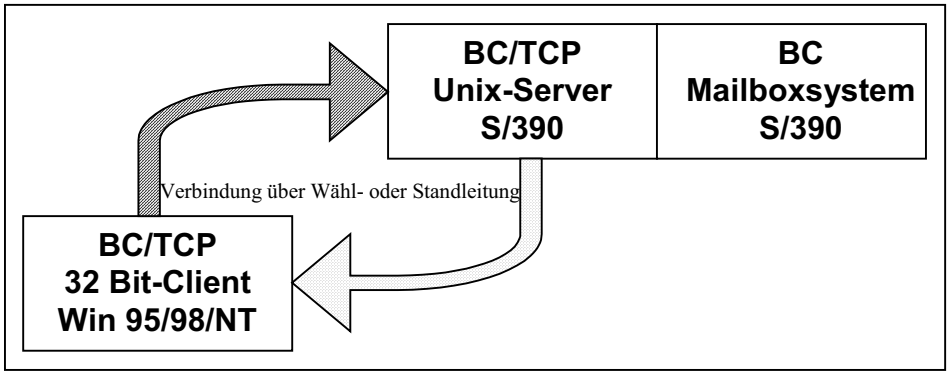

Abbildung 1: BusinessContact-System mit Windows-Client

Der Austausch von Daten (z.B. EDIFACT-Daten) erfolgt vom **Windows-Client** (auf einem PC) über eine Wähl- oder Standleitung zu einem **UNIX-Server** (auf einem S/390), der wiederum die Kommunikation mit dem **Mailbox-Host** (auf dem selben S/390) durchführt.

2. Installation

Die Systemvoraussetzungen für einen ordnungsgemäßen Betrieb der BC/TCP-Software sind

- Windows 95/98/NT
- Netzwerkzugang (entweder Standleitung oder Wählzugang)
- für Wählzugang:
  - Modem
  - AT&T-Netzwerk-Registrierung (ehemals IBM Global Network)
- ca. 5 MB freier Plattenspeicher

Falls Sie keinen Wählleitungszugang, sondern eine Standleitung besitzen, so überspringen Sie die folgenden Kapitel bis zum Punkt **Prüfung der Verbindung**.

2.1. Wählzugang für Windows installieren

Die in diesem Kapitel angeführten Installationsanweisungen beziehen sich auf die deutsche Version von Windows 95.

2.1.1. Hinweise

Für Windows 98 und Windows NT gelten die selben Installationsregeln, allerdings ist die Vorgehensweise auf diesen Systemen etwas unterschiedlich zu der bei Windows 95. Bei anderssprachigen Versionen ist natürlich auf die entsprechenden Sprachunterschiede zu achten.

Es kann auch durch verschiedene Versionen des Betriebssystems zu leicht unterschiedlichen Meldungen kommen. Betriebssystem-Upgrades stehen bei Microsoft zum Download bereit. Upgrades für Ihr Windows-Betriebssystem erhalten Sie unter http://windowsupdate.microsoft.com.

2.1.2. Installationsanweisungen für Windows 95

Sie müssen folgende Soft- und Hardware auf Ihrem Computer installiert haben:

- Modem
- DFÜ-Netzwerk
- DFÜ-Adapter und TCP/IP-Protokoll

2.1.3. Modem installieren

Folgen Sie den mit Ihrem Modem mitgelieferten Installationsanweisungen für Windows 95 und gehen Sie sicher, daß Ihr Modem auch ordnungsgemäß funktioniert.

2.1.4. DFÜ-Netzwerk installieren

Wählen Sie **Start** Ą **Einstellungen** Ą **Systemsteuerung**. Doppelklicken Sie **Software** und wählen Sie **Windows-Setup** aus. Klicken Sie auf **Verbindungen** in der Liste **Komponenten** und drücken Sie auf **Details**. Vergewissern Sie sich, daß die Option **DFÜ-Netzwerk** ausgewählt ist.

Wenn dies der Fall ist, gehen Sie zum nächsten Punkt, ansonsten wählen Sie es aus und drücken Sie auf **OK**. Windows wird nun versuchen, die notwendigen Treiber von den Installationsdisketten oder der CD-ROM zu installieren.

Nun ist das DFÜ-Netzwerk installiert.

2.1.5. DFÜ-Adapter und TCP/IP-Protokoll installieren

Wählen Sie **Start** Ą **Einstellungen** Ą **Systemsteuerung**. Doppelklicken Sie **Netzwerk**. Vergewissern Sie sich, daß **Konfiguration** ausgewählt ist. Sowohl der **DFÜ-Adapter** als auch **TCP/IP** sollten vorhanden sein.

Wenn dies der Fall ist, gehen Sie zum nächsten Punkt.

Um den DFÜ-Adapter zu installieren, drücken Sie **Hinzufügen**, doppelklicken Sie **Netzwerkkarte**, scrollen Sie dann in der Herstellerliste zu **Microsoft** und wählen Sie **DFÜ-Adapter** aus. Drücken Sie dann OK.

Wenn Sie das TCP/IP-Protokoll installieren müssen, drücken Sie **Hinzufügen**, doppelklicken Sie **Protokoll**, scrollen Sie dann in der Herstellerliste zu **Microsoft** und wählen Sie **TCP/IP** aus. Drücken Sie dann **OK**.

Nun sollte Ihre Konfiguration sowohl den **DFÜ-Adapter** als auch das **TCP/IP-Protokoll** beinhalten.

Wählen Sie den **DFÜ-Adapter** aus, drücken Sie auf **Eigenschaften**, wählen Sie **Bindungen** und überprüfen Sie, ob **TCP/IP** ausgewählt ist. Dürcken Sie **OK**, um das Eigenschaftsfenster zu verlassen und nochmals **OK**, um aus den Netzwerkeinstellungen zu kommen.

Windows muß nun restartet werden, damit die Änderungen in Kraft treten.

2.2. Konfiguration der Wählzugangssoftware

Nachdem Sie die nötige Software installiert haben, müssen Sie sie richtig konfigurieren, um sich zum AT&T Global Network (früher IBM Global Network) verbinden zu können.

2.2.1. DFÜ-Netzwerk konfigurieren

Sie müssen eine neue DFÜ-Netzwerkverbindung erstellen, um Zugang zum AT&T Global Network zu bekommen.
Wählen Sie dafür **Start** Ą **Programme** Ą **Zubehör** Ą **DFÜ-Netzwerk** und doppelklicken Sie **Neue Verbindung erstellen**.

Geben Sie der Verbindung einen Namen (z.B. ATT Global Network) und wählen Sie ein Modem. Merken Sie sich diesen Namen, denn Sie werden ihn für den Betrieb von BC/TCP benötigen. Drücken Sie den **Weiter-Button**.

Geben Sie nun die Telefonnummer für Ihren Zugang ein. In der Regel wird dies **071891** (Ortskennzahl) **2145** (Rufnummer) und **Österreich (43)** (Landeskennzahl) sein. Mit dieser Nummer benutzen Sie den österreichweiten Online-Tarif. Spezielle Einwählnummern finden Sie unter http://www.attbusiness.net/hlpctr/accessnum.html. Betätigen Sie den **Weiter-Button** und drücken Sie **Fertigstellen**.

2.2.2. TCP/IP-Protokoll konfigurieren

Wählen Sie **Start** Ą **Einstellungen** Ą **Systemsteuerung** und doppelklicken Sie **Netzwerk**. Wählen Sie **TCP/IP** Ą **DFÜ-Adapter** aus und drücken Sie dann auf **Eigenschaften**. Sie sehen nun das Eigenschaftsfenster von TCP/IP.

Konfigurieren Sie die sechs Sektionen wie beschrieben:

- **IP-Adresse:**
  Wählen Sie **IP-Adresse automatisch beziehen**
- **WINS-Konfiguration:**
  Wählen Sie **WINS-Auflösung deaktivieren** (Sie können es später wieder aktivieren)
- **Gateway:**
  Diese Sektion muß leer bleiben.
- **Bindungen:**
  Defaultmäßig ist **Client für Microsoft-Netzwerke** eingestellt. Ändern Sie das nicht.
- **Erweitert:**
  Ändern Sie in dieser Sektion nichts.
- **DNS-Konfiguration:**
  Wählen Sie **DNS deaktivieren**.

Drücken Sie nun zweimal **OK**, um die Netzwerkeinstellungen zu verlassen. Windows muß restartet werden, damit die Änderungen in Kraft treten.

2.2.3. Besonderheiten bei der Installation unter Windows 98

Das DFÜ-Netzwerk befindet sich bei Windows 98 unter dem Menüpunkt **Start** Ą **Programme** Ą **Zubehör** Ą **Kommunikation** Ą **DFÜ-Netzwerk**.

2.2.4. Besonderheiten bei der Installation unter Windows NT

Bei Windows NT 4.0 erfolgt die Installation des DFÜ-Netzwerkes etwas anders. Wenn Sie eine neue DFÜ-Verbindung erstellen, so müssen folgende Werte ausgewählt werden:

- **Eine Verbindung mit dem Internet wird hergestellt**.
- **Der Nicht-Windows NT-Server erwartet die Eingabe von entweder Anmeldeinformationen ...**

- **Eintrags- und Modemeigenschaften bearbeiten Ą Sicherheit Ą Beliebige Echtheitsbestätigung (einschl. unverschlüsselte) annehmen**
- **Das Feld Domain bei der Einwahl muss leer bleiben!**

2.3. Prüfung der Verbindung

Falls Sie einen Wählleitungszugang haben, so erstellen Sie mit dem DFÜ-Netzwerk eine Verbindung mit dem AT&T Global Network. Wählen Sie dafür aus dem DFÜ-Netzwerk Ihre erstellte Verbindung aus und starten Sie diese mit einem Doppelklick. Sie werden aufgefordert, Ihren Benutzernamen und Ihr Kennwort einzugeben.

In das Feld Benutzername geben Sie Ihren AT&T Global Network-Zugangsaccount und die User-ID in der Form **SECUREIP.Account.UserID** ein. Hat z.B. der Kunde ***Toll AG*** von der IBM den Zugangsaccount ***ATTOL*** und die UserID ***AT$TOL01*** erhalten, so müßte er in das Feld Benutzername ***SECUREIP.ATTOL.AT$TOL01*** eingeben.

Als Kennwort geben Sie das Ihnen mitgeteilte Passwort ein. Bedenken Sie, daß nach einigen Fehlversuchen mit einem falschen Passwort Ihr Zugang gesperrt wird.

Sie können Ihr Passwort ändern, indem Sie es in der Form "altesPasswort.neuesPasswort" (ohne Hochkomma) eingeben. Beim Verbindungstest empfiehlt es sich, das Ihnen bekanntgegebene Passwort sofort zu ändern.

Betätigen Sie den **Verbinden-Button** und überprüfen Sie, ob Ihre Wählverbindung funktioniert. Windows gibt dabei in einem Fenster Statusnachrichten über die Verbindung aus.

Sie haben nun eine Verbindung zum AT&T Global Network über eine Wähl- oder Standleitung.

Überprüfen Sie nun aus einem MS-DOS-Fenster mit folgendem Befehl, ob der BC/TCP-Server erreichbar ist.

ping 141.95.210.220

Die TCP/IP-Adresse des BC/TCP-Servers ist 141.95.210.220.

Wenn Sie eine Antwort vom Server erhalten, dann sind Sie bereit, BC/TCP ordnungsgemäß zu starten. Wenn bei der Überprüfung ein Fehler auftritt, so liegt ein Problem in der Konfiguration Ihrer Netzwerksoftware vor. Überprüfen Sie, ob Sie die Windows-Konfiguration wie oben beschrieben durchgeführt haben, wenn ja, überprüfen Sie, ob Sie Ihr Modem und Ihren Wählleitungszugang bzw. Ihren Standleitungszugang ordnungsgemäß eingerichtet haben.

Nach dem Test sollen Sie sich, falls Sie sich mit einem Modem eingewählt haben, wieder disconnecten.

2.4. Installation von BC/TCP

BC/TCP können Sie sich vom Internet als ein Installationspaket herunterladen. Starten Sie das Setup-Programm. Das Programm leitet Sie durch die Installation.

2.5. Upgrade von vorhergehenden Versionen

Falls Sie BC/TCP bereits auf Ihrem Rechner installiert haben und auf die aktuelle Version upgraden wollen, so sichern Sie bitte zuerst die Datei BcTcp.ini aus dem BC/TCP-Programmverzeichnis.
Deinstallieren Sie dann aus der Windows-Systemsteuerung (Software) die installierte Version von BC/TCP.
Installieren Sie dann die aktuelle Version und überschreiben Sie die neu installierte Datei BcTcp.ini mit der zuvor gesicherten INI-Datei.

3. BC/TCP-Bedienung

3.1. User Interface

Der BC/TCP-Win32-Client bietet Ihnen ein komfortables, modernes User Interface, mit dem Sie Ihre Transaktionen zum BusinessContact-Mailboxsystem einfach durchführen können.

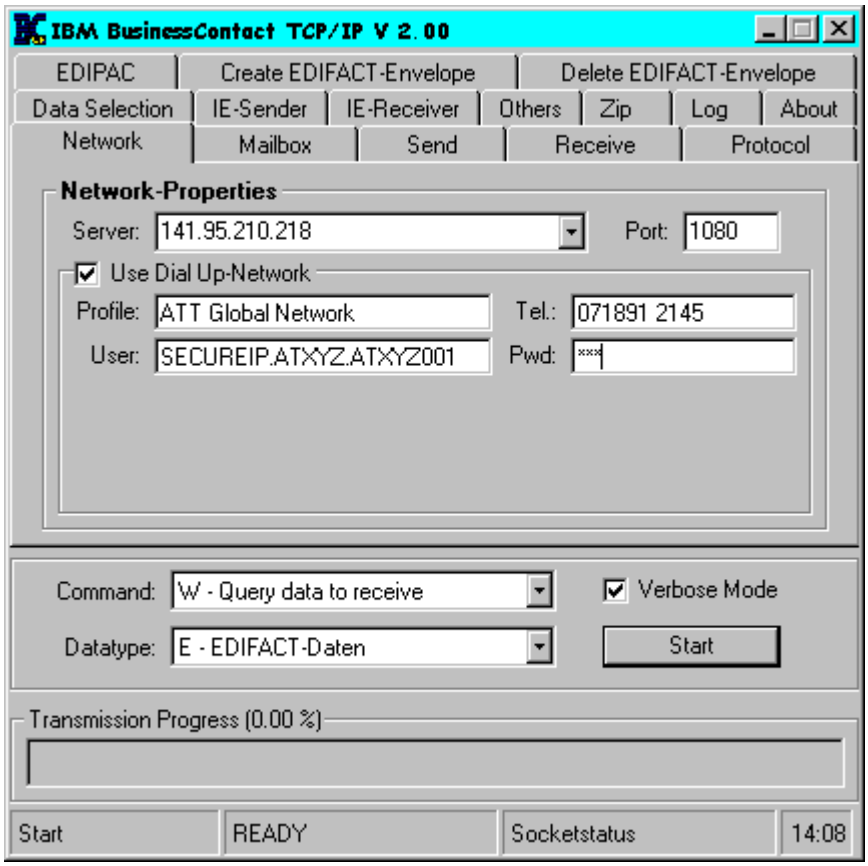

Abbildung 2: BC/TCP-User Interface

Neben dem User Interface besitzen Sie aber auch die Möglichkeit, BC/TCP über eine Befehlsdatei zu steuern (Details dazu siehe weiter unten).

Die graphische Benutzeroberfläche ist in 3 Hauptteile gegliedert:

Der **obere Bereich** dient zur Eingabe von Daten für die Steuerung der Kommunikation.

Die Daten sind nach bestimmten Kategorien geordnet. Zu einem Zeitpunkt kann jeweils nur eine Kategorie angezeigt werden, Sie können aber durch die Auswahl

eines Registerelements (z.B. Network, Mailbox, usw.) die entsprechende Kategorie anzeigen und bearbeiten.

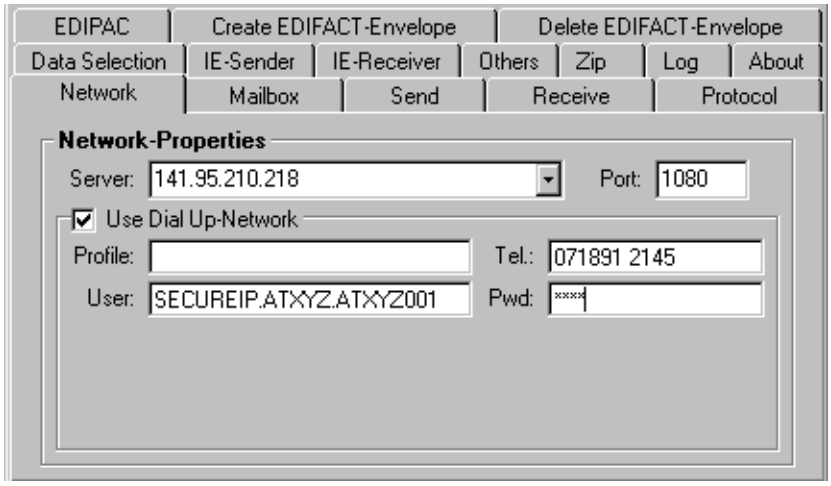

Abbildung 3: Oberer Bereich

Im **mittleren Bereich** können Sie die Transaktion auswählen und starten.

Sie müssen das Kommando (z.B. Senden, Empfangen, Reaktivieren, Löschen, usw.) im Feld **Command** und den Datentyp der Transaktion (z.B. EDIFACT, SEDAS, usw.) im Feld **Datatype** einstellen. Sie können die Bildschirmausgaben durch Deaktivieren der Checkbox **Verbose Mode** ausschalten. Durch Betätigen des **Start-Buttons** starten Sie die Transaktion. Nach erfolgter Transaktion wird – sofern der **Verbose Mode** eingeschaltet ist - das Datenaustauschprotokoll im Fileviewer angezeigt, der durch Betätigen des **OK-Buttons** wieder verlassen werden kann.

Abbildung 4: Mittlerer Bereich

Der **untere Bereich** dient für Status- und Fortschrittsangaben während der Kommunikation.

Es wird ein Balken angezeigt, der den prozentuellen Fortschritt einer Dateiübertragung anzeigt. Direkt darunter befinden sich drei Felder für Statusangaben, wobei das linke Feld die gerade durchgeführte Aktion anzeigt (z.B. Transfer), das mittlere Feld den Status (z.B. Sending oder Fehlernachrichten) und das rechte Feld den Status der TCP-Verbindung (z.B. Closed). Die Uhrzeit wird rechts unten angezeigt.

Nach dem Ende einer Transaktion wird im Statusfeld im Fehlerfall eine Fehlermeldung (ERROR_xxx, wobei xxx dem Clientstatus entspricht, als der Fehler auftrat) oder READY im Fall einer ordnungsgemäß beendeten Transaktion angezeigt.

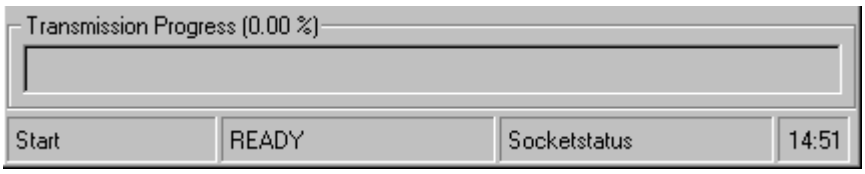

Abbildung 5: Unterer Teil

3.2. Kommunikationsparameter

Neben dem Kommando und dem Datentyp gibt es eine Reihe von Parametern für eine Transaktion, die Sie im oberen Teil des User Interfaces einstellen können.

Diese Parameter sind nach Kategorien geordnet. Diese können durch Drücken auf das entsprechende Registerelement (z.B. Network, Mailbox, usw.) angezeigt und in weiterer Folge bearbeitet werden.

3.2.1. Network-Properties

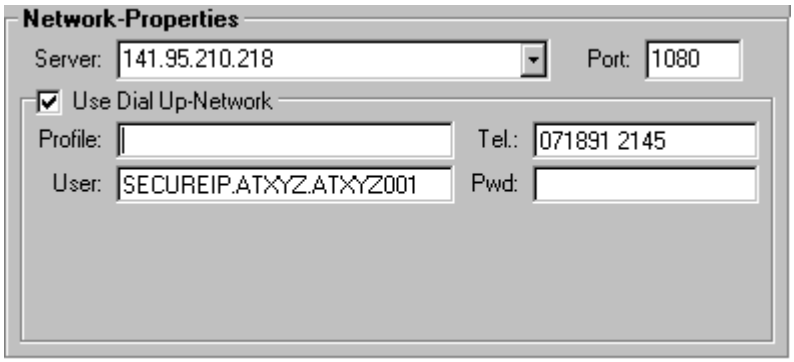

Abbildung 6: Network-Properties

Die Network-Properties legen die Einstellungen für die Verbindung zum BC/TCP-Server fest. Da diese Einstellungen nur einmal durchgeführt werden müssen, empfiehlt es sich, sie in der Initialisierungsdatei zu speichern (siehe Kapitel **Initialisierungs- und Befehlsdatei**).

Die Parameter **Server** und **Port** müssen auf die Werte **141.95.210.220** (die IP-Adresse des Servers) und **1080** (der Port für das BusinessContact-Kommunikationsprogramm) gesetzt werden.

Wenn Sie die Verbindung zum AT&T Global Network über eine Wählverbindung erstellen, so aktivieren Sie die Checkbox **Use Dial Up-Network**.

Im Feld **Profile** geben Sie dann den Namen der Verbindung ein, die Sie im DFÜ-Netzwerk konfiguriert haben. Falls Sie in der Konfiguration dieser Verbindung die Telefonnummer, den Benutzeraccount und das Passwort nicht gespeichert haben, so müssen Sie diese Werte in den Feldern **Nr.**, **User** und **Pwd.** eingeben, wobei das Feld **User** die Form **SECUREIP.Account.UserID** aufweisen muß. Account und UserID werden Ihnen von IBM mitgeteilt (wenn z.B. der Account **ATTOL** und die UserID **AT$TOL01** ist, so muß dieser Eintrag **SECUREIP.ATTOL.AT$TOL01** lauten).

Sie können Ihr Passwort ändern, indem Sie im Feld **Pwd.** **altesPasswort.neuesPasswort** eingeben. Soll z.B. das Passwort von **hello** auf **world** geändert werden, so müßte in das Feld **Pwd. hello.world** eingegeben werden.

3.2.2. Mailbox-Properties

Abbildung 7: Mailbox-Properties

Die Mailbox-Properties legen die Einstellungen für das Anmelden im BusinessContact-Mailboxsystem fest. Geben Sie in die Felder **User-ID** und **Password** die entsprechenden Werte ein und wählen Sie die Checkbox **Use Testsystem**, wenn Sie auf das Testsystem und nicht das Produktionssystem zugreifen wollen.

Die Zugangskennung zum BusinessContact-Mailboxsystem unterscheidet sich von der Zugangskennung zum AT&T Global Network. Erstere besteht aus einer einfachen User-ID in der Form **$XXXnn**, wobei **XXX** eine Abkürzung für den Kundennamen ist und **nn** eine laufende Nummer (z.B. $IBM01). Zweitere wird an Benutzer von Wählleitungen vergeben und setzt sich aus einem Account und einer User-ID zusammen (Beschreibung siehe Kapitel **Installation**).

Da auch diese Einstellungen selten geändert werden, empfiehlt es sich, sie in der Initialisierungsdatei zu speichern, wobei zu beachten ist, daß das Passwort in dieser Datei unverschlüsselt abgespeichert wird und jeder, der Zugriff auf diese Datei hat, auch Zugriff auf Ihr Passwort hat (siehe Kapitel **Initialisierungs- und Befehlsdatei**).

Wenn Sie Ihr Passwort ändern wollen, so wählen Sie die Checkbox **Change Password** aus und geben Sie das neue Passwort zweimal in die Felder **New Password** und **Confirm Password** ein.

3.2.3. Send-Properties (für Cmd. Send)

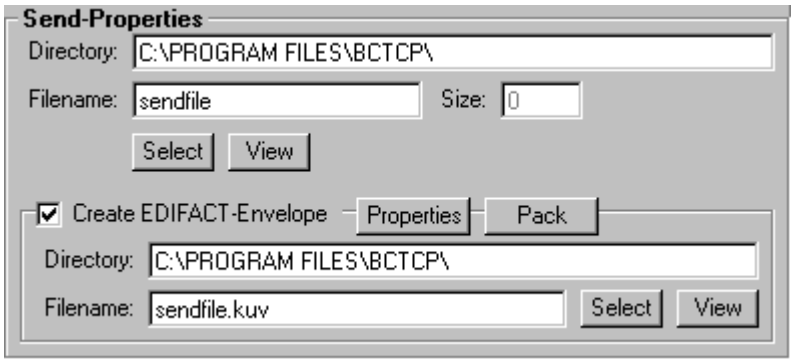

Abbildung 8: Send-Properties

Für das Senden von Daten müssen die Einstellungen in dieser Kategorie durchgeführt werden. Sie können entweder eine Datei händisch in die Felder **Directory** und **Filename** eingeben, es empfiehlt sich allerdings, die zu sendende Datei über den File-Dialog auszuwählen, der durch das Betätigen des **Select File-Buttons** geöffnet wird. Mit dem **View File-Button** kann die Datei im Fileviewer angezeigt werden.

Neu ab Version 2.00 ist das automatische Kuvertieren von Dateien ("Packen"). Damit ist es möglich, um Daten, die noch kein EDIFACT-Kuvert haben, ein solches zu erstellen. Wählen Sie dafür die Checkbox **Create EDIFACT-Envelope** aus und geben Sie den Namen, wie die kuvertierte Datei heissen soll, in die Felder **Directory** bzw. **Filename** ein. Sie können diese Datei - falls sie schon existiert - mit dem **Select-Button** auswählen oder z.B. nach der Übertragung mit dem **View-Button** im Fileviewer anzeigen.

Weitere Einstellungen für die Erstellung des EDIFACT-Kuverts finden Sie unter den **Create EDIFACT-Envelope-Properties**, zu denen Sie durch Betätigung des **Properties-Button** direkt wechseln können.

Das Kuvertieren findet nur dann statt, wenn Sie den **Start-Button** betätigen und Sie den **Befehl Send** und den **Datentyp EDIFACT** ausgewählt haben.

Durch Drücken des **Pack-Buttons** wird die Kuvertierung alleine durchgeführt, d.h., die Daten werden dabei nicht versendet. Sie können sich das Ergebnis der Kuvertierung durch den **View-Button** im Fileviewer ansehen.

3.2.4. Receive-Properties (für Cmd. Receive, Reactivate)

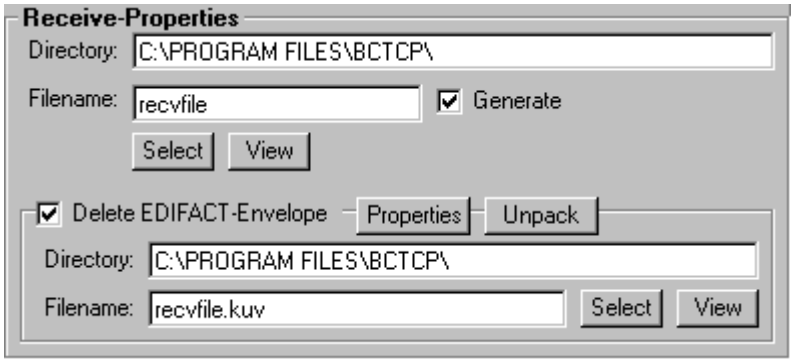

Abbildung 9: Receive-Properties

Sollen Daten vom Mailbox-System empfangen werden, so müssen in dieser Kategorie die entsprechenden Daten eingegeben werden. Im Feld **Directory** wird das Verzeichnis gespeichert, in dem die empfangene Datei abgespeichert wird. Für die Vergabe des Dateinamens haben Sie drei Möglichkeiten, wobei die beiden erstgenannten durch eine Deaktivierung der Checkbox **Generate Name** ermöglicht wird:

Auswahl über den Fileselect-Dialog, der durch Betätigen des **Select File-Buttons** aufgerufen werden kann. Die ausgewählte Datei wird dabei von der abgeholten Datei überschrieben.
Manuelle Eingabe eines Dateinamens.
Automatische Generierung des Dateinamens durch das System. Dieser Dateiname besitzt die Form „**F**JJMMTThhmmssttt“ (16 Zeichen), wobei die Buchstaben hinter dem F dem aktuellen Datum und der aktuellen Zeit entsprechen. Diese Option kann durch Auswählen der Checkbox **Generate Name** eingeschaltet werden. Etwaige Daten im Feld **Filename** werden ignoriert.

Neu ab Version 2.00 ist das automatische Dekuvertieren von Dateien ("Entpacken"). Damit ist es möglich, von Daten, die ein EDIFACT-Kuvert haben, dieses zu

entfernen. Wählen Sie dafür die Checkbox **Delete EDIFACT-Envelope** aus und geben Sie den Namen, wie die dekuvertierte Datei heissen soll, in die Felder **Directory** bzw. **Filename** ein. Sie können diese Datei - falls sie schon existiert - mit dem **Select-Button** auswählen oder z.B. nach der Übertragung mit dem **View-Button** im Fileviewer anzeigen.

Weitere Einstellungen für die Erstellung des EDIFACT-Kuverts finden Sie unter den **Delete EDIFACT-Envelope-Properties**, zu denen Sie durch Betätigung des **Properties-Button** direkt wechseln können.

Das Dekuvertieren findet nur dann statt, wenn Sie den **Start-Button** betätigen und Sie den **Befehl Receive/Reactivate** und den **Datentyp EDIFACT** ausgewählt haben.

Durch Drücken des **Unpack-Buttons** wird die Dekuvertierung alleine durchgeführt, d.h., die Daten werden dabei nicht empfangen. Sie können sich das Ergebnis der Dekuvertierung durch den **View-Button** im Fileviewer ansehen.

3.2.5. Protocol-Properties (für alle Cmd.)

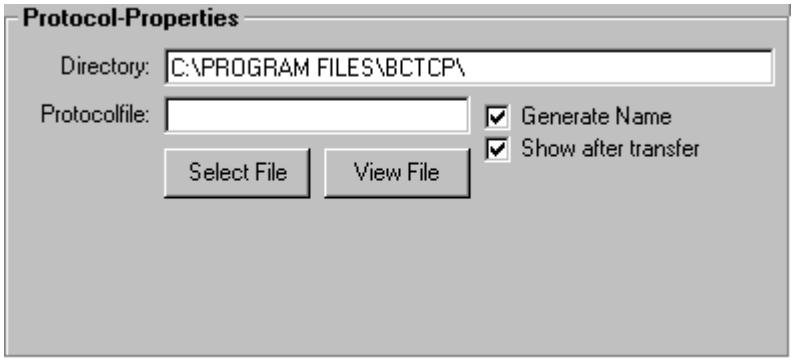

Abbildung 10: Protocol-Properties

Für jede Transaktion wird ein Datenaustauschprotokoll erstellt. Dieses wird am Client gespeichert, und die Einstellungen dafür können in dieser Kategorie vorgenommen werden.

Sie haben die gleichen Möglichkeiten wie bei den Receive-Properties (siehe oben), zusätzlich können Sie aber durch das Auswählen der Checkbox **Show after transfer** festlegen, ob das Datenaustauschprotokoll nach dem Empfang im Fileviewer angezeigt wird.

Der automatisch generierte Protokollname beginnt mit einem P, sonst unterscheidet er sich nicht vom automatisch generiertem Receive-Filenamen.

3.2.6. EDIPAC-Properties

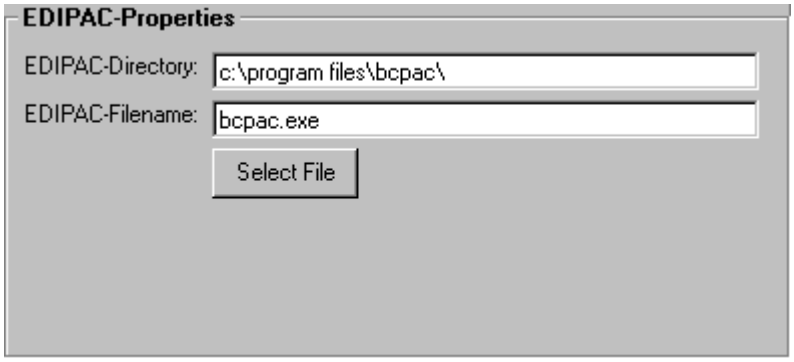

Abbildung 11: EDIPAC-Properties

Die automatische Kuvertierung und Dekuvertierung wird von einem von IBM entwickelten Programm durchgeführt. Dieses Programm (BCPAC oder EDIPAC) muss auf Ihrem Computer installiert sein.

Hier geben Sie Pfad und Dateiname des EDIPAC-Programmes in den Parametern **EDIPAC-Directory** bzw. **EDIPAC-Filename** an. Sie können die Datei auch über den Select File-Button auswählen.

Wenn Sie eine neue Version von EDIPAC installieren, brauchen Sie hier nur den Pfad und den Dateinamen zu ändern (sofern dies nicht mit den bisherigen Werten übereinstimmt).

Da diese beiden Parameter sich nicht oft ändern, empfiehlt es sich, diese Parameter im Ini-File zu speichern.

3.2.7. Properties for creating an EDIFACT-Envelope

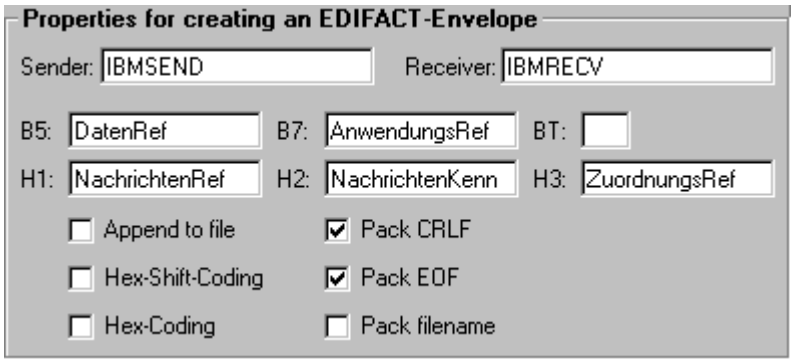

Abbildung 12 : Properties for creating an EDIFACT-Envelope

Wenn um zu versendende Daten ein EDIFACT-Kuvert automatisch erzeugt werden soll, so müssen hier die für dieses Kuvert notwendigen Parameter eingegeben werden.

Die Mindestangaben sind die Felder **Sender** und **Receiver**, bei denen es sich um die Mailboxnamen des Absenders (also Ihnen) und des Empfängers (also Ihres Datenaustauschpartners) handelt.

Zusätzlich können Sie noch verschiedene Parameter des EDIFACT-Kuverts in den restlichen Feldern dieser Maske angeben. Um eine genauere Beschreibung dieser Felder zu erhalten, bewegen Sie den Mauszeiger über eines der Felder und warten Sie, bis der Hilfetext erscheint. Dort ist dann eine detailliertere Information über das entsprechende Feld ersichtlich.

3.2.8. Properties for deleting the EDIFACT-Envelope

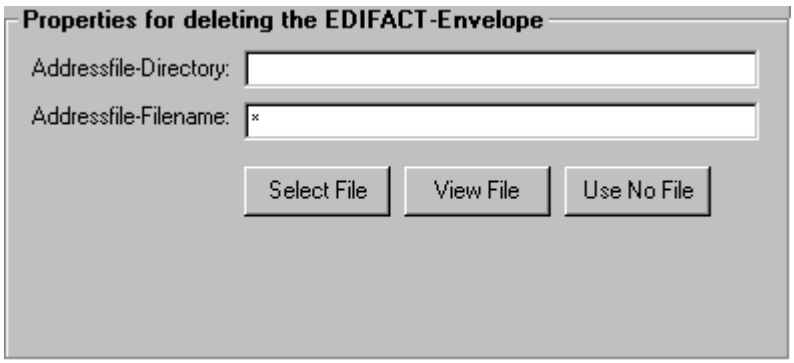

Abbildung 13: Properties for deleting the EDIFACT-Envelope

Sie können beim automatischen Entfernen des EDIFACT-Kuverts die Aufsplittung der in der Datei enthaltenen EDIFACT-Nachrichten über ein sogenanntes Adressfile regeln. Dieses Adressfile wird durch die Felder **Addressfile-Directory** und

Addressfile-Filename definiert. Sie können dieses Addressfile auch über den **Select File-Button** auswählen und mit dem **View File-Button** im Fileviewer anzeigen.

Genauere Details zur Funktion und zum Aufbau eines Addressfiles entnehmen Sie bitte der Dokumentation Ihres EDIPAC-Programmes.

Wenn Sie keine Aufsplittung über ein Adressfile durchführen wollen, so können Sie dies durch Betätigen des **Use No File-Buttons** definieren. Das Feld **Addressfile-Directory** bleibt dann leer und das Feld **Addressfile-Filename** wird auf "*" gesetzt.

3.2.9. Selection-Properties

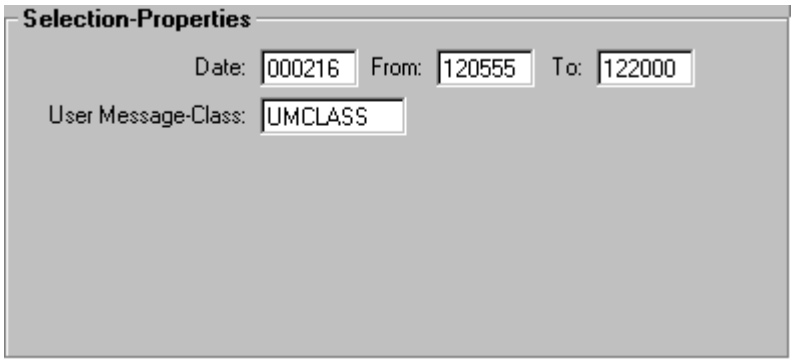

Abbildung 14: Selection-Properties

Für das Reaktivieren und Holen von Daten können Selektionen auf die Daten durchgeführt werden, wobei im Feld **Date** die Daten nach dem Datum in der Form JJMMTT selektiert werden können, die Felder **From** (nur in Verbindung mit **Date**) und **To** (nur in Verbindung mit **From**) beziehen sich auf die laufende Nummer der einzelnen Sendungen und begrenzen einen Bereich.

Im Feld **UserMessage-Class** können Daten nach ihrer User Message Class klassifiziert werden. Damit ist es möglich, verschiedene Arten von EDIFACT-Nachrichten zu unterscheiden.

3.2.10. InformationExchange-Sender

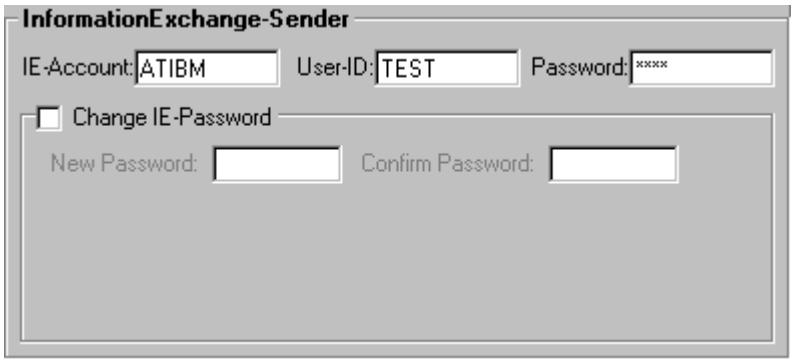

Abbildung 15: InformationExchange-Sender

Beim Datenaustausch mit InformationExchange, dem internationalen IBM Mailboxsystem, müssen hier die entsprechenden Einstellungen durchgeführt werden. In die Felder **IE-Accout**, **User-ID** und **Password** müssen Sie **Ihre** InformationExchange-spezifischen Eingaben machen, das Passwort kann analog zum Passwortändern für BusinessContact (siehe Login-Properties) geändert werden.

3.2.11. InformationExchange-Receiver

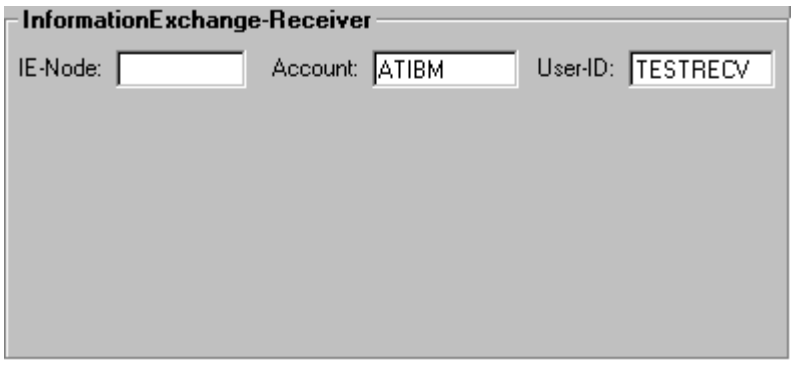

Abbildung 16: InformationExchange-Receiver

Wenn Daten vom Datentyp **sonstige Daten** an einen IE-Benutzer geschickt werden, so müssen in den Feldern **Node**, **Account** und **User-ID** die entsprechenden Daten des Empfängers angegeben werden. Das Feld **Node** kann dabei leer bleiben, wenn es sich um einen europäischen Empfänger handelt.

3.2.12. Other Properties

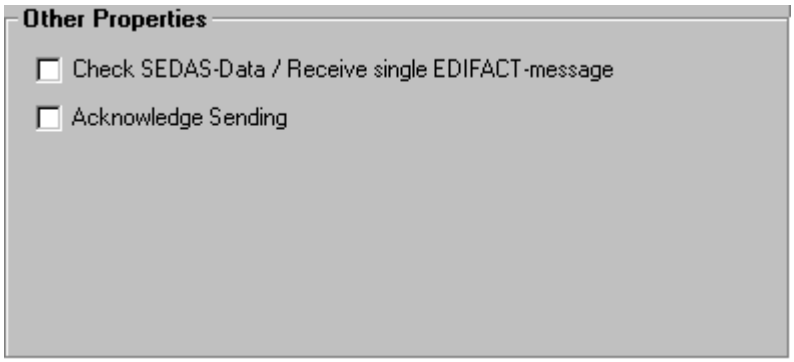

Abbildung 17: Other Properties

In dieser Kategorie können folgende Zusatzoptionen eingestellt werden:

Die Checkbox **Check SEDAS-Data** bestimmt, ob Daten vom Typ **SEDAS-Bestelldaten** geprüft werden sollen, wenn es sich um ein Kommando mit dem Datentyp SEDAS handelt (entspricht „Extra Parameter“ = „P“).
Neu ab Version 1.3: Beim Abholen von EDIFACT-Daten bewirkt das Einschalten dieses Parameters, dass immer nur eine Sendung (die Daten innerhalb eines EDIFACT-Kuverts) abgeholt wird (entspricht **Extra Parameter** = „1“).
Diese Checkbox existiert nur mehr aus Kompatibilitätsgründen und sollte ab Version 2.20 nicht mehr verwendet werden, da er durch das Feld **Extra Parameter** wesentlich erweitert wird.

Neu ab Version 2.20: Das Feld **Extra Parameter** (ohne Abbildung) besteht aus einem Zeichen, das unverändert zum Server gesendet wird, um damit zukünftige Serviceerweiterungen nutzbar zu machen. Dieses Feld wird von der Verwendung der Checkbox **Check Sedas-Data** überschrieben, man sollte also nur einen dieser beiden Parameter verwenden!
Aktuelle Extra-Serviceerweiterungen:
Wert ‚P‘: SEDAS-Bestelldaten werden geprüft (entspricht aktiviertem CHKSEDAS-Flag)
Wert ‚1‘: Edifact-Daten werden einzeln abgeholt (entspricht aktiviertem CHKSEDAS-Flag)
Wert ‚U‘: Bei den Kommandos W (query data to receive) und F (query sent data) bewirkt dieser Wert, dass das Datenaustauschprotokoll nur nicht-abgeholte Datensendungen anzeigt. Anstatt der Sendezeit scheint die User Message Class im Datenaustauschprotokoll auf. Bereits abgeholte Sendungen scheinen nicht im Protokoll auf.

Die Checkbox **Acknowledge Sending** bestimmt, ob eine Empfangsbestätigung vom Empfänger angefordert werden soll.

3.2.13. Zip-Properties

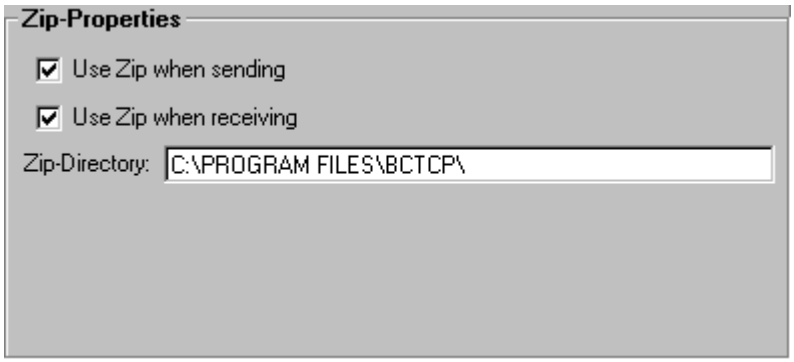

Abbildung 18: Zip-Properties

BC/TCP bietet Ihnen die Möglichkeit, durch den Einsatz moderner Zip-Routinen die Daten für die Übertragung zu komprimieren. Dadurch wird die Übertragungszeit wesentlich verringert. Es ist allerdings anzumerken, daß das Zippen erst ab einer Dateigröße von ca. 1000 Byte relevante Komprimierungsraten schafft und somit einen Vorteil bringt.

Die Zip-Optionen können in der Kategorie Zip-Properties eingestellt werden. Der Benutzer kann mit den Checkboxen **Use Zip when sending** und **Use Zip when receiving** wählen, ob das Zippen beim Senden und/oder beim Empfangen von Daten (und damit auch beim Empfangen des Protokolls) verwendet wird. Der Pfad für das Zipprogramme zip.exe ist defaultmäßig das Programmverzeichnis, weil diese Programmdateien bei der Installation von BC/TCP in diesem Verzeichnis installiert wird. Dieser Pfad wird im Feld **Zip-Directory** angegeben.

Es handelt sich in der vorliegenden Version um die InfoZip-Programme, die unter http://www.cdrom.com/pub/infozip frei erhältlich sind.

3.2.14. Protocollog-Properties

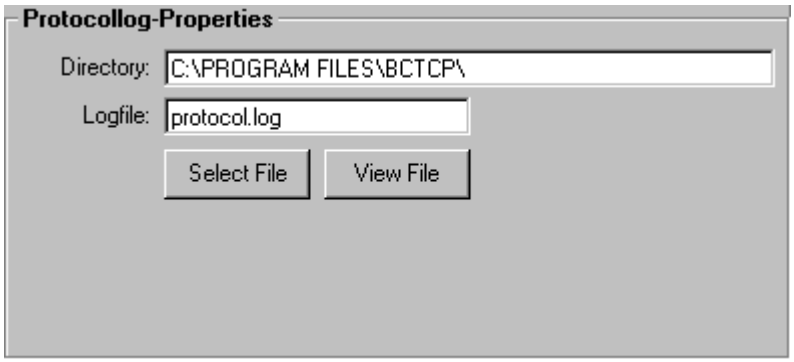

Abbildung 19: Protocollog-Properties

Die Datenaustauschprotokolle werden chronologisch sortiert in einem chronologischen Protokolllog gespeichert, der in der hier angeführten Datei gespeichert wird.

Überschreitet diese Datei eine bestimmte Größe, so wird die Datei auf den angeführten Namen plus eine Extension mit einer aufsteigenden Nummer umkopiert und eine neue Datei mit diesem Namen erzeugt.

3.2.15. About

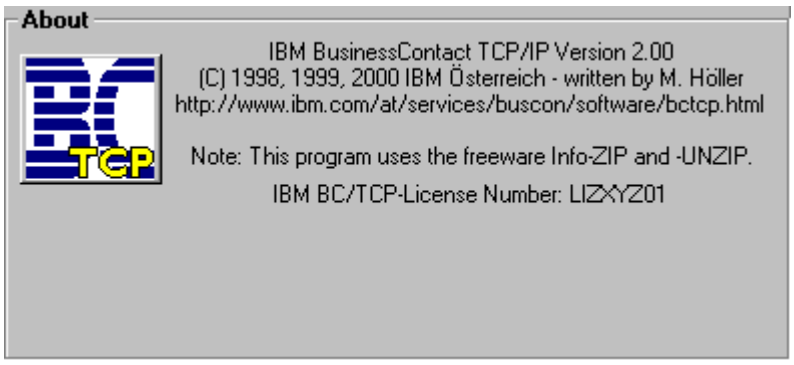
About
IBM BusinessContact TCP/IP Version 2.00
(C) 1998, 1999, 2000 IBM Österreich - written by M. Höller
http://www.ibm.com/at/services/buscon/software/bctcp.html
Note: This program uses the freeware Info-ZIP and -UNZIP.
IBM BC/TCP-License Number: LIZXYZ01

Abbildung 20: About

In dieser Kategorie erhalten Sie Informationen über die vorliegende Version von BC/TCP wie z.B. die Lizenznummer.

Durch einen Mausklick auf das IBM BC/TCP-Logo wird der Programmlog angezeigt, der im Fehlerfalle nützliche Informationen zur Fehlerbehebung geben kann.

Überschreitet dieser Programmlog (BcTcp.log) eine bestimmte Größe, so wird die Datei auf den angeführten Namen plus eine Extension mit einer aufsteigenden

Nummer umkopiert und eine neue Datei mit diesem Namen erzeugt (z.B. BcTcp.log.1, BcTcp.log.2, usw.).

3.3. Test von BC/TCP

Um BC/TCP und die Parametereinstellungen zu testen, gehen Sie sicher, daß in den **Connection-Properties** die Parameter **Server** und **Port** richtig gesetzt sind und – falls Sie eine Wählleitung benutzen – die Einstellungen der entsprechenden Parameter **Profile**, **Nr.**, **User** und **Pwd.**.

Setzen Sie die Parameter **Loginname**, **Passwort** und **Use Testsystem** in den **Login-Properties** auf die richtigen Werte und führen Sie den Command **W – Query data to receive** aus.

Wenn alles richtig funktioniert, wird nach der Transaktion das Datenaustauschprotokoll im Fileviewer angezeigt, sofern Sie den **Verbose Mode** eingeschaltet haben.

Im Fehlerfall erhalten Sie eine entsprechende Nachricht, Sie können darüberhinaus auch noch den Log überprüfen.

4. Automatische Steuerung

4.1. Initialisierungsdatei

Alle Parameter von BC/TCP können in einer Initialisierungsdatei gespeichert werden. Diese Datei befindet sich im Programmverzeichnis und heißt „**BcTcp.ini**“. Da es sich dabei um eine gewöhnliche Textdatei handelt, kann sie mit einem normalen Editor bearbeitet werden.

Die meisten Eintragungen in dieser Datei entsprechen einem Eingabefeld im User Interface, d.h., die INI-Datei wird gelesen und die Werte der Parameter in die entsprechenden Felder gestellt, so als ob sie direkt eingegeben worden wären. Dem Benutzer bleibt also erspart, gleichbleibende Parameter – wie z.B. den Servername – immer wieder neu eingeben zu müssen.

Nach dem Einlesen der INI-Datei beim Programmstart wird – im Gegensatz zu einer Befehlsdatei - keine Transaktion gestartet.

4.2. Befehlsdatei

Der Benutzer hat die Möglichkeit, in der INI-Datei im Parameter **EXECFILE** eine Befehlsdatei (oder Commandfile) anzugeben, der sofort nach dem Programmstart abgearbeitet und ausgeführt wird. Die Befehlsdatei hat das gleiche Format wie die INI-Datei, es können jedoch in diesem Commandfile auch mehrere Transaktionen, die sequentiell ausgeführt werden, angeführt sein.

Die Parameter werden zuerst aus der INI-Datei gelesen, und dann der Reihe nach aus dem Commandfile, wobei nach jedem aus dem Commandfile eingelesenen Befehl eine Transaktion gestartet wird.

Wichtig dabei ist, daß zuvor eingelesene Parameter nicht zurückgesetzt werden, wenn sie nicht überschrieben werden, d.h., daß z.B. eine Einstellung in der INI-Datei für alle Transfers im Commandfile gilt, sofern diese Einstellung nicht explizit im Commandfile anders gesetzt wird.

Sind in einer Befehlsdatei mehrere Kommandos angeführt und wird die Abarbeitung nicht von einem Benutzer kontrolliert, so empfiehlt es sich, den **Verbose Mode**

(Parameter **VERBOSEM**) auszuschalten, da das Programm sonst Fehlermeldungen oder Protokolle am Bildschirm anzeigt, die vom Benutzer bestätigt werden müßten. Dadurch würde das Programm nicht mit der Verarbeitung fortfahren.

Die Ergebnisse der Abarbeitung der Befehlsdatei werden in einer Datei gespeichert, die den gleichen Dateinamen wie die Befehlsdatei mit der Extension **„.OUT“** aufweist und im Programmverzeichnis gespeichert wird.

Existiert eine solche Datei bereits, so wird diese nicht gelöscht, sondern die neuen Verarbeitungsergebnisse werden an diese Datei angefügt.

Heißt die Befehlsdatei also z.B. *buson.txt*, so werden die Ergebnisse in der Datei *buscon.txt.OUT* gespeichert. Darüberhinaus werden alle Ergebnisse im Programmlog protokolliert.

In dieser Textdatei wird der Status der Abarbeitung in folgender Formatierung ausgegeben, wobei für jeden Befehl genau eine Zeile in dieser Datei eingetragen wird:

Stellen	**Länge**	**Beschreibung**
1 – 2	2	In diesen zwei Zeichen wird die **Reihenfolge** der Befehle bei der Befehlsdateiabarbeitung angezeigt (z.B. *01* für den ersten Befehl in der Datei, *02* für den zweiten, usw.).
3	1	leer
4 – 18	15	**Datum** in der Form *„JJMMTT hh:mm:ss“*
19	1	leer
20 – 35	16	**Befehlsbeschreibung** (z.B. *SEND* oder *RECEIVE*, usw.)
36	1	leer
37 – 68	32	**Status** des Befehls (z.B. *READY*, falls die Verarbeitung OK war, oder *ERROR_LOGIN* für einen Login-Fehler. Fehler beginnen immer mit *„ERROR_“*).
69	1	leer
70 – 85	16	**Dateiname** der abgeholten Datei (bei RECEIVE oder REACTIVATE) bzw. der Protokolldatei (bei allen übrigen Befehlen)

Das Feld Status kann einen von folgenden Werten annehmen: READY, ABNORMAL_ERROR, ERROR_PARAM, ERROR_CONNECT, ERROR_LOGIN, ERROR_TRANSFER, ERROR_CHECKSUM, ERROR_SENDCOMMIT, ERROR_WAITFORPROTOCOL, ERROR_PROTOCOL, ERROR_WAITFORDATA, ERROR_DATA, ERROR_ZIP, ERROR_UNZIP

4.3. Programmparameter

Neben der Möglichkeit, die Abarbeitung einer Befehlsdatei in der Initialisierungsdatei zu bestimmen, hat der Benutzer die Möglichkeit, eine Abarbeitung einer Befehlsdatei über die Angabe eines Programmparameters zu erzwingen.

Als Parameter muß dabei die Befehlsdatei angegeben werden.

Ist ein solcher Parameter vorhanden, so wird zuerst die INI-Datei eingelesen, dann – falls vorhanden - die in der INI-Datei angegebene Befehlsdatei ausgeführt, dann wird die als Parameter angegebene Befehlsdatei ausgeführt und dann wird das Programm wieder beendet.

Diese Variante eignet sich besonders für die Steuerung von BC/TCP von einem anderen Programm aus. Als Ergebnis der Abarbeitung muß die für jede Befehlsdatei automatisch generierte OUT-Datei (siehe oben) abgefragt werden.

Wird kein Parameter angegeben, so wird BC/TCP nicht automatisch beendet.

4.4. Einbindung von BC/TCP in andere Programme

BC/TCP kann, da es in Form einer DLL ausgeliefert wird, sehr einfach in andere Applikationen eingebunden werden. Folgendes Beispiel zeigt die Einbindung in ein Visual Basic-Programm.

```
Private BcTcp As clBcTcp      'Variable für Klasseninstanz
Private rc As String          'Variable für Rückgabewert
Private cmdFile As String     'Variable für Commandfile

Public Sub CallBcTcpDLL()
  Set BcTcp = New clBcTcp     'Erzeugung einer Instanz dieser Klasse
  rc = BcTcp.Start(cmdFile)   'Aufruf der Methode Start, als Para-
                              'meter kann ein abzuarbeitender
                              'Commandfile angegeben werden
  Set BcTcp = Nothing         'Freigabe der Instanz
End Sub
```

Um die DLL verwenden zu können, muß ein Verweis darauf erzeugt werden (in Visual Basic kann die über das Menü Projekt Ą Verweise gemacht werden). Die DLL heißt BcTcp.dll und befindet sich im Programmverzeichnis von BC/TCP.

Der Rückgabewert der Methode Start ist ein String, in dem der Status der Verarbeitung gespeichert ist (siehe Beschreibung der OUT-Datei). Um die Abarbeitung eines Commandfiles zu überprüfen, sollte das entsprechende OUT-File überprüft werden (Beschreibung siehe oben).

4.5. Aufbau der Befehlsdatei

Ein Commandfile besteht aus folgenden Teilen:

[<Befehlskennung>]
<Parameterkennung>(<Parameterwert>)
<Parameterkennung>(<Parameterwert>)
[<Befehlskennung>]
<Parameterkennung>(<Parameterwert>)
<Parameterkennung>(<Parameterwert>)
<Parameterkennung>(<Parameterwert>)
usw.

Anmerkungen:

- Eine Befehlskennung muß genau einem Befehl entsprechen (Auflistung der Befehle siehe unten). Kommen zwischen den []-Klammern zusätzliche Zeichen vor, so wird der Befehl ignoriert.
- In einem Commandfile können mehrere Befehlssequenzen vorkommen. Diese werden dann nacheinander abgearbeitet.
- Die Parameterkennung ist 8 Zeichen lang.
- Zeichen vor der Parameterkennung werden ignoriert.
- Die Parameterkennung ist NICHT case sensitiv.
- Zwischen der Parameterkennung und der (-Klammer darf sich kein Zeichen, also auch kein Blank befinden, sonst wird dieser Parameter ignoriert. Um einen Parameter auszukommentieren, kann z.B. zwischen Parameterkennung und (-Klammer ein !-Zeichen eingefügt werden.
- Der Parameterwert ist beliebig lang und wird (mit Ausnahme von Flags) case sensitiv übertragen.
- Parameterkennungen, die ein Flag darstellen, besitzen einen boolschen Parameterwert. Dabei steht eine Parameterwert von “y” oder “Y” für Ja oder True, alles andere für Nein oder False.

Beispiel für Einträge in eine INI- oder Commanddatei:

```
[query]                                Query-Kommando
datatype(e)                            EDIFACT-Daten
use_dial(y)                            Verbindung über Windows Dial Up Net
dialentr(ATT Global Network)           Telefonbucheintrag im DUN
dialteln(071891 2145)                  Telefonnummer für das DUN
dialuser(secureip.atibm.at$ibm01)      Account.Benutzernummer für das DUN
dialpass(dummypass)                    Password für das DUN
bc_login!($xyz01)                      Login (wird ignoriert - Rufzeichen)
bc_passw (passwd)                      Pwd (wird ignoriert - Blank)
use_test(Y)                            Testsystem benützen
sendpath(d:\bctcp\outfiles)            Verzeichnisse für Out-...
recvpath(d:\bctcp\infiles)             ...und In-Dateien
autorecv(y)                            Name automatisch festlegen
protpath(d:\bctcp\prot\)               Verzeichnis für Protokoll
autoprot(Y)                            Name automatisch festlegen
zip_path(d:\bctcp)                     Verzeichnis für Zip-Prg.
log_path(d:\bctcp\prot)                Verzeichnis für Protokolllog
log_file(d:\bctcp\protocol.log)        Name der Protokolllogdatei
bclicens(LIZXYZ01)                     Lizenznummer
execfile!(d:\bctcp\cmdfile.in)         Commandfile (wird ignoriert)
verbosem(y)                            Meldungen anzeigen
```

4.6. Befehle

Mögliche Werte für die Befehlskennung sind:

SEND: Senden von Daten.

RECEIVE: Empfangen von Daten.

REACTIVATE: Reaktivieren von Daten.

DELETE: Löschen von Daten.

QSENT: Abfrage von gesendeten Daten.

QUERY: Abfrage von zum Empfang bereiten Daten.

4.7. Parameter

4.7.1. Transferparameter

DATATYPE:

Zeigt den Typ der Daten an. Dabei steht **E** für EDIFACT-Daten, **A** für

Abrechnungsdaten, **B** für SEDAS-Bestelldaten, **R** für SEDAS-Rechnungsdaten, **D** für Verteilerlisten und **S** für sonstige Daten.

4.7.2. Verbindungsparameter

- **BCSERVER: BusinessContact-Server**
  Name des Servers, auf dem das BC/TCP-Serverprogramm läuft. Zur Zeit ist dies der Rechner mit der IP-Adresse '141.95.210.220'

- **BCPORTNO: BusinessContact-Portnumber**
  Nummer des Ports, über den eine Verbindung zum BC/TCP-Serverprogramm erstellt werden kann. Zur Zeit ist dies der Port 1080.

- **USE_DIAL: Flag: Use Dial Up Network**
  Dieses Flag zeigt an, ob das Windows Dial Up Network (DUN) verwendet wird, um eine Verbindung zum Server herzustellen. Dies ist dann der Fall, wenn der Kunde nicht über eine Standleitung zum AT&T Global Network verfügt und der Zugang dazu über ein Modem erfolgt.

- **DIALENTR: Dial Up Network-Telephonebook Entry**
  Dieser Wert legt den im DFÜ-Netzwerk festgelegten Eintrag (die Verbindung) fest, der für die Wählverbindung verwendet wird.

- **DIALTELN: Dial Up Network-Telephone Number**
  Diese Nummer wird verwendet, um eine Wählverbindung herzustellen. Sie muß nicht angeführt werden, wenn im Telephonebook Entry die entsprechende Nummer gespeichert ist.

- **DIALUSER: Dial Up Network-User ID**
  Die Benutzerkennung für das AT&T Global Network (zumeist in der Form ATIBM.ATIBM001).

- **DIALPASS: Dial Up Network-Password**
  Das Passwort für den Wählzugang zum AT&T Global Network.

4.7.3. Loginparameter

- **BC_LOGIN: BusinessContact-Loginname**
  Username des BusinessContact-Benutzers.

- **BC_PASSW: BusinessContact-Password**
  Passwort des BusinessContact-Benutzers. VORSICHT: Falls Sie das Passwort in

einem Commandfile abspeichern, müssen Sie sich bewußt sein, daß jeder, der Zugriff auf den Commandfile hat, Ihr Passwort lesen kann.

- **USE_TEST: Flag: Use Testsystem**
  Dieses Flag legt fest, ob das BusinessContact-Testsystem oder das Produktionssystem verwendet wird.

4.7.4. Sendeparameter

- **SENDPATH: Path of outgoing Datafiles**
  Verzeichnis, wo sich die zu sendende Datei befindet (z.B. 'C:\BCTCP\OUTFILES\'). Das Verzeichnis muß existieren, sonst wird das Programmverzeichnis verwendet.

- **SENDFILE: Name of outgoing Datafile**
  Name der zu sendenden Datei. Diese muß sich in Verzeichnis SENDPATH befinden.

4.7.5. Empfangsparameter

- **RECVPATH: Path of incoming Datafiles**
  Verzeichnis, wo die empfangene Datei gespeichert werden soll. (z.B. 'C:\BCTCP\INFILES\'). Das Verzeichnis muß existieren, sonst wird das Programmverzeichnis verwendet.

- **RECVFILE: Name of incoming Datafile**
  Die Datei wird nach dem Empfang vom Server im Verzeichnis RECVPATH unter diesem Namen abgespeichert, falls AUTORECV nicht gesetzt ist. Existiert bereits eine Datei mit einem solchen Namen, so wird ein neuer Name automatisch generiert.

- **AUTORECV: Flag: Generate Name of incoming Datafile**
  Dieses Flag zeigt an, daß die empfangene Datei nicht wie im Parameter RECVFILE angeführt genannt wird, sondern daß der Name vom System automatisch generiert wird (z.B. **F**980701153000123, wobei die Zahlen das Format JJMMTThhmmssttt aufweisen).

4.7.6. Parameter zum Empfangen des Protokolls

∉# **PROTPATH: Path of Protocol**

Verzeichnis, wo das empfangene Datenaustauschprotokoll gespeichert werden soll.

∉# **PROTFILE: Name of Protocol**

Das Datenaustauschprotokoll wird nach dem Empfang vom Server im Verzeichnis PROTPATH unter diesem Namen abgespeichert, falls AUTOPROT nicht den Wert 'Y' aufweist. Existiert bereits eine Datei mit einem solchen Namen, so wird ein neuer Name automatisch generiert.

∉# **AUTOPROT: Flag: Generate Name of Protocol**

Dieses Flag zeigt an, daß das empfangene Datenaustauschprotokoll nicht wie im Parameter PROTFILE angeführt genannt wird, sondern daß der Name vom System automatisch generiert wird (z.B. **P**980701153000123, wobei die Zahlen das Format JJMMTThhmmssttt aufweisen). Sind beim Empfangen von Daten sowohl AUTORECV als auch AUTOPROT gesetzt, so unterscheiden sich die Namen des Datenfiles und des Datenaustauschprotokolls nur im ersten Buchstaben (F bzw. P).

∉# **SHOWPROT: Flag: Show Protocol after Transmission**

Dieses Flag zeigt an, ob das Datenaustauschprotokoll nach dem Empfang im Fileviewer angezeigt wird. Diese Anzeigen muß durch Drücken des OK-Buttons im Fileviewer beendet werden, deshalb sollte dieses Flag im Batch-Betrieb ausgeschaltet sein.

4.7.7. EDIPAC-Parameter

∉# **EDIPPATH: Path of EDIPAC-Program**

Dieser Parameter gibt das Verzeichnis an, wo das EDIPAC-Programm (BCPAC) installiert wurde.

∉# **EDIPFILE: Name of EDIPAC-Program**

Dieser Parameter gibt den Namen (ohne Verzeichnis) an, wie das EDIPAC-Programm (BCPAC) heisst.

4.7.8. Parameter für die automatische Kuvertierung

∉# **CREDIENV: Flag: Create EDIFACT-Envelope**

Dieses Flag zeigt an, ob die zu versendende Datei automatisch kuvertiert wird,

d.h., ob um die Daten aus der Sendedatei ein EDIFACT-Kuvert automatisch erstellt wird.

∉# **PACKPATH: Path of packed File**

Dieser Parameter gibt das Verzeichnis an, wo die kuvertierte Datei gespeichert werden soll.

∉# **PACKFILE: Name of packed File**

Dieser Parameter gibt den Namen an, unter dem die kuvertierte Datei gespeichert werden soll.

∉# **PACKSNDR: Mailbox of Sender (ECODEX: GLN)**

Dieser Parameter gibt die Mailbox des Absenders an (z.B. ATTOLL oder 9001234567890).

∉# **PACKRCVR: Mailbox of Receiver (ECODEX: GLN)**

Dieser Parameter gibt die Mailbox des Empfängers an (z.B. ATTOLL oder 9001234567890).

∉# **PACKUNB5: Datenaustauschreferenz**

Dieser Parameter gibt den Wert des UNB-Segmentes/Feld 5 an. Dabei handelt es sich um die Datenaustauschreferenz.

∉# **PACKUNB7: Anwendungsreferenz**

Dieser Parameter gibt den Wert des UNB-Segmentes/Feld 7 an. Dabei handelt es sich um die Anwendungsreferenz (oder User Message Class).

∉# **PACKUNBT: Testkennzeichen**

Dieser Parameter gibt ein einstelliges Testkennzeichen im UNB-Segment an.

∉# **PACKUNH1: Nachrichtenreferenz**

Dieser Parameter gibt den Wert des UNH-Segmentes/Feld 1 an. Dabei handelt es sich um die Nachrichtenreferenz.

∉# **PACKUNH2: Nachrichtenkennung**

Dieser Parameter gibt den Wert des UNH-Segmentes/Feld 2 an. Dabei handelt es sich um die Nachrichtenkennung.

∉# **PACKUNH3: Zuordnungsreferenz**

Dieser Parameter gibt den Wert des UNH-Segmentes/Feld 3 an. Dabei handelt es sich um die Zuordnungsreferenz.

∉# **PACKAPPD: Flag: Append do File**

Dieses Flag zeigt an, ob die Daten an eine bestehende Datei angehängt werden.

- **PACKHSCD: Flag: Process special Hex-Shift-Coding**
  Dieses Flag zeigt an, ob beim Packen ein spezieller Hex-Shift-Coding-Algorithmus angewendet werden soll.

- **PACKHXCD: Flag: Process special Hex-Coding**
  Dieses Flag zeigt an, ob beim Packen ein spezieller Hex-Coding-Algorithmus angewendet werden soll.

- **PACKCRLF: Flag: Pack CarriageReturn/LineFeed**
  Dieses Flag zeigt an, ob CRLF-Zeichen mitgepackt werden.

- **PACKEOFC: Flag: Pack EndOfFile-Character**
  Dieses Flag zeigt an, ob das EOF-Zeichen mitgepackt wird.

- **PACKFNAM: Flag: Pack Filename**
  Dieses Flag zeigt an, ob der Dateiname mitgepackt wird.

4.7.9. Parameter für die automatische Dekuvertierung

- **DLEDIENV: Flag: Delete EDIFACT-Envelope**
  Dieses Flag zeigt an, ob die empfangenen oder reaktivierten Daten automatisch dekuvertiert werden, d.h., ob das EDIFACT-Kuvert von den empfangenen oder reaktivierten Daten automatisch entfernt wird.

- **UNPKPATH: Path of unpacked File**
  Dieser Parameter gibt das Verzeichnis an, in dem die entpackte Datei gespeichert wird.

- **UNPKFILE: Name of unpacked File**
  Dieser Parameter gibt den Namen an, unter dem die entpackte Datie gespeichert wird.

- **ADDRPATH: Path of Addressfile**
  Dieser Parameter gibt das Verzeichnis an, in dem die Adressdatei gespeichert ist.

- **ADDRFILE: Name of Addressfile**
  Dieser Parameter gibt den Namen an, unter dem die Adressdatei gespeichert ist.

4.7.10. Parameter für die Datenselektion

- **SEL_DATE: Date for Selection**
  Mit diesem Datum haben Sie die Möglichkeit, Daten eines bestimmten Tages zu

bearbeiten. Das Datum muß in der Form JJMMTT angegeben werden und bezieht sich immer auf das Sendedatum des Datenpaketes.

- **SEL_FRNO: From-Number for Selection**
  Ist die im Datenaustauschprotokoll unter lfd. Nummer angegebene Paketnummer. Die Eingabe wirkt nur in Verbindung mit dem Sendedatum.

- **SEL_TONO: To-Number for Selection**
  Soll ein geschlossener Block von Datenpaketen bearbeitet werden, so kann dies durch Angabe von SEL_DATE und SEL_FRNO des ersten gewünschten Datenpakets und Eintragung von SEL_TONO (inklusive) erreicht werden. Hier gilt die Reihung im Datenaustauschprotokoll.

- **SEL_UMCL: User Message Class**
  Hier kann der Benutzer die gesendeten oder zu holenden Daten klassifizieren.

4.7.11. Parameter für InformationExchange (Sender)

Diese Parameter sind nur dann für den internationalen Datenaustausch erforderlich, wenn der Datensender eine eigene IE Mailbox besitzt.

- **IEX_ACNT: InformationExchange-Account ID**
  Die im internationalen Service vergebene Verrechnungsnummer des Datensenders. Wird diese nicht angegeben so wird die bei IBM gespeicherte IE Account Id angenommen.

- **IEX_USER: InformationExchange-User ID**
  Die Benutzernummer im internationalen Service. Bleibt dieses Feld blank dann wird die bei IBM gespeicherte IE Benutzernummer angenommen.

- **IEX_PASS: InformationExchange-Password**
  Das aktuelle Password im internationalen Service. Wird dieser Parameter angegeben und das Kommando Receive angegeben, so wird die internationale Mailbox ausgelesen und die Daten mit den nationalen Daten an den Abrufer übermittelt.

4.7.12. Parameter für InformationExchange (Receiver)

Diese Parameter sind nur erforderlich, wenn das Kommando **Send** mit dem Datentyp **Sonstigen Daten** ausgewählt wurde.

⋐# **MBX_NODE: International Mailbox-Name**
Der Systemname der internationalen Mailbox. Dieser Parameter muß bei nicht europäischen Partnern angegeben werden.

⋐# **MBX_ACNT: International Mailbox-Account ID**
Die im internationalen Service vergebene Verrechnungsnummer des Datenempfängers.

⋐# **MBX_USER: International Mailbox-User ID**
Die im internationalen Service definierte Benutzernummer des Empfängers.

4.7.13. Spezialparameter

⋐# **CHKSEDAS: Flag: Check SEDAS-Data / Receive single EDIFACT-message**
Dieses Flag zeigt an, ob SEDAS Bestelldaten durch eine Vollprüfung laufen (entspricht EXTRAPAR='P').
Wenn Sie EDIFACT-Daten abholen wollen, bewirkt das Einschalten dieses Flags, dass jeweils nur **eine** EDIFACT-Nachricht abgeholt wird, also die Daten innerhalb eines EDIFACT-Kuverts (entspricht EXTRAPAR='1').
Dieser Parameter existiert nur mehr aus Kompatibilitätsgründen und sollte nicht mehr verwendet werden. Stattdessen sollte der Parameter EXTRAPAR (siehe unten) benutzt werden.

⋐# **EXTRAPAR: Extra Parameter**
Dieser Parameter (ab V.2.20) besteht aus einem Zeichen, das unverändert zum Server gesendet wird, um damit zukünftige Serviceerweiterungen nutzbar zu machen. **Dieser Parameter wird von der Verwendung des CHKSEDAS-Parameters überschrieben, man sollte also nur einen dieser beiden Parameter verwenden!**
Aktuelle Extra-Serviceerweiterungen:
Wert ‚P': SEDAS-Bestelldaten werden geprüft (entspricht aktiviertem CHKSEDAS-Flag)
Wert ‚1': Edifact-Daten werden einzeln abgeholt (entspricht aktiviertem CHKSEDAS-Flag)
Wert ‚U': Bei den Kommandos W (query data to receive) und F (query sent data) bewirkt dieser Wert, dass das Datenaustauschprotokoll nur nicht-abgeholte Datensendungen anzeigt. Anstatt der Sendezeit scheint die User Message Class im Datenaustauschprotokoll auf. Bereits abgeholte Sendungen scheinen nicht im Protokoll auf.

⋐# **ACK_SEND: Flag: Acknowledge Data sent to IE**
Eine Empfangsbestätigung (Acknowledgement) wird dann mit diesem Flag

angefordert, wenn die Sendung durch den Empfänger bestätigt werden soll (nur für IE).

4.7.14. Parameter für Datenkompression

- **ZIP_SEND: Flag: Use Zip when Sending**
  Dieses Flag zeigt an, ob Daten, die vom Client zum Server geschickt werden, komprimiert werden.

- **ZIP_RECV: Use Zip when Receiving**
  Dieses Flag zeigt an, ob Daten, die vom Server zum Client geschickt werden, komprimiert werden.

- **ZIP_PATH: Path of InfoZip- and InfoUnzip-Program**
  Dieser Parameter beinhaltet den Pfad, wo sich das InfoZip-Programm befindet. Dieses Programm wird mit BC/TCP ausgeliefert und befindet sich nach der Installation im selben Verzeichnis wie das Hauptprogramm.

4.7.15. Parameter für den Protokolllog

- **LOG_PATH: Path of Protocollog**
  Die Datenaustauschprotokolle werden historisch gespeichert. Diese Logdatei befindet sich in diesem Verzeichnis, welches existieren muß, sonst wird das Programmverzeichnis gewählt.

- **LOG_FILE: Name of Protocollog**
  Die Logdatei hat diesen Namen.

4.7.16. Lizenzparameter

- **BCLICENS: BusinessContact-Licensenumber**
  Von IBM vergebene, 8-stellige Lizenznummer. Die Vergabe der Lizenznummern wird bei der Auslieferung von BC/TCP geregelt.

4.7.17. Programmparameter

- **EXECFILE: Execute Commandfile**
  (nur für Initialisierungsdatei) Dieser Parameter gibt einen Commandfile an, der nach dem Laden der Initialisierungsdatei abgearbeitet wird.

- **VERBOSEM: Verbose Mode**

  Dieses Flag zeigt an, ob Datenaustauschprotokolle (falls SHOWPROT gesetzt ist) und Fehlermeldungen am Bildschirm angezeigt werden.

- **CONNTIME: Timeout for Connecting**

  Dieser numerische Wert legt die Zeitdauer in Sekunden fest, die – bei bestehender Wähl- oder Standleitungsverbindung – auf das Herstellen der Sockets-Kommunikation wartet. Die Voreinstellung ist 15 Sekunden.

5. Protokoll des BC/TCP-Gateways [1]

Die Kommunikation zwischen dem Client und dem BC/TCP-Server funktioniert über eine TCP/IP-Sockets-Verbindung über den Port 1080.

Je nach Art der Transaktion ist zwischen verschiedenen Protokollen zu unterscheiden, die in der Folge angeführt sind.

5.1. Protokoll für Löschen und Abfragen

Server-Status	**Data In**	**Data Out**
Wait for connection (accept)	(socket-connect)	
Wait for login	**BCLIN** (5 B) <commandString> (200 B)	
Commit login data		**BCLIN** (5 B)
Mailbox-Transmission		(login is sent to mailbox-host) (protocol is received from mailbox-host) [client is waiting]
Send protocol-command		**BCSFL** (5 B) <filename> (16 B) <filesize> (16 B) <sendExtraParameter> (16 B)
Send protocol-data		<data> (<filesize> B)
Send protocol-checksum		**BCFLE** (5 B) <checksum> (16 B)
Wait for protocol-commit/abort	**BCCOM** oder **BCABR** oder BCRET (5B)	
Disconnect		(socket-disconnect)

[1] Die in diesem Kapitel angeführten Informationen sind nur für die Implementierung weiterer BC/TCP-Clients wichtig.

5.2. Protokoll für Senden

Server-Status	**Data In**	**Data Out**
Wait for connection (accept)	(socket-connect)	
Wait for login	**BCLIN** (5 B) <commandString> (200 B)	
Commit login data		**BCLIN** (5 B)
Receive data	<data> (length in commandStr)	
Send checksum		**BCFLE** (5 B) <checksum> (16 B)
Wait for commit/abort	**BCCOM** (commit) oder **BCABR** (abort) oder BCRET (retry) (5 B)	
Mailbox-Transmission		(login is sent to mailbox-host) (data is sent to mailbox-host) (protocol is received from mailbox-host) [client is waiting]
Send protocol-command		**BCSFL** (5 B) <filename> (16 B) <filesize> (16 B) <sendExtraParameter> (16 B)
Send protocol-data		<data> (<filesize> B)
Send protocol-checksum		**BCFLE** (5 B) <checksum> (16 B)
Wait for protocol-commit/abort	**BCCOM** oder **BCABR** oder BCRET (5B)	
Disconnect		(socket-disconnect)

5.3. Protokoll für Holen und Reaktivieren

Server-Status	Data In	Data Out
Wait for connection (accept)	(socket-connect)	
Wait for login	**BCLIN** (5 B) <commandString> (200 B)	
Commit login data		**BCLIN** (5 B)
Mailbox-Transmission		(login is sent to mailbox-host) (protocol is received from mailbox-host) [client is waiting]
Send protocol-command		**BCSFL** (5 B) <filename> (16 B) <filesize> (16 B) <sendExtraParameter> (16 B)
Send protocol-data		<data> (<filesize> B)
Send protocol-checksum		**BCFLE** (5 B) <checksum> (16 B)
Wait for protocol-commit/abort	**BCCOM** oder **BCABR** oder BCRET (5B)	
Mailbox-Transmission		(data is received from mailbox-host) [client is waiting]
Send data-command		**BCSFL** (5 B) <filename> (16 B) <filesize> (16 B) <sendExtraParameter> (16 B)
Send data		<data> (<filesize> B)
Send data-checksum		**BCFLE** (5 B) <checksum> (16 B)
Wait for data-commit/abort	**BCCOM** oder **BCABR** oder BCRET (5B)	
Disconnect		(socket-disconnect)

5.4. Erläuterungen zum Protokoll

- Die Checksummenprüfung muß für Send UND Receive am Client erfolgen, wobei die Checksumme (Typ double in C) die Summe aller einzelnen Byte-Werte der Datei ist.
- Retry für eine Dateiübertragung ist noch nicht unterstützt
- Jeder Parameter (außer der commandString) ist 16 Byte, jeder Befehl 5 Byte lang.
- Anstatt der Kommandos/Daten, die vom Server (!) gesendet werden, kann auch folgende Fehlernachricht verschickt werden, wobei xxx (3 B) für die Länge der Message steht:

  BCERRxxx<ErrorMessage>

  Nach dem Senden dieser Nachricht wird die Socketverbindung geschlossen!

Sollte z.B. während der Übertragung zum Mailbox-Host ein Fehler auftreten, so wird der Client davon auf diese Art und Weise benachrichtigt.

- Auf alle socket-read und –write besteht auf der Serverseite ein Timeout von z.Zt. 30 Sekunden. Danach wird die Verbindung abgebaut.
- **Commandstring**:

  Stellen 1 - 110: siehe allgemeine ECODEX-Beschreibung

  Stelle 111: ‘T’ für Testsystem

  Stelle 112: ‘Z’ für Zip beim Hochladen vom Client zum Server

  Stelle 113: ‘Z’ für Zip beim Runterladen vom Server zum Client

  Stellen 114 - 118: reserviert

  Stellen 119 - 134: Filename

  Stellen 135 - 150: Filesize

  Stellen 151 - 158: von IBM vergebene Client-Info (z.B. „BTP02.20” für Windows Client 1.1)

  Stellen 159 - 166: von IBM vergebene Lizenznummer

  Stellen 167 – 182: Empfängername im datakom-Gateway

  Stellen 183 - 200: reserviert
- **sendExtraParameter**:

  Stelle 1: Datentyp (z.B. ‘E’ für EDIFACT)

  Stelle 2: ‘Z’ für gezippte Datei

  Stellen 3 - 16: reserviert

5.5. Client-Status

In der Statusausgabe des User Interfaces und in den Programmlog werden – je nach Status – Meldungen angezeigt, die Auskunft darüber geben, in welchem Zustand sich der Client gerade befindet. So wird, wenn der Client z.B. auf das Datenaustauschprotokoll wartet, die Meldung **Waiting for protocol...** ausgegeben. Beim Senden einer Datei erscheint während der Dateiübertragung die Meldung **Sending data...**.

Der Win32-Client weist während der Kommunikation mit dem BC/TCP-Server folgende Stati auf:

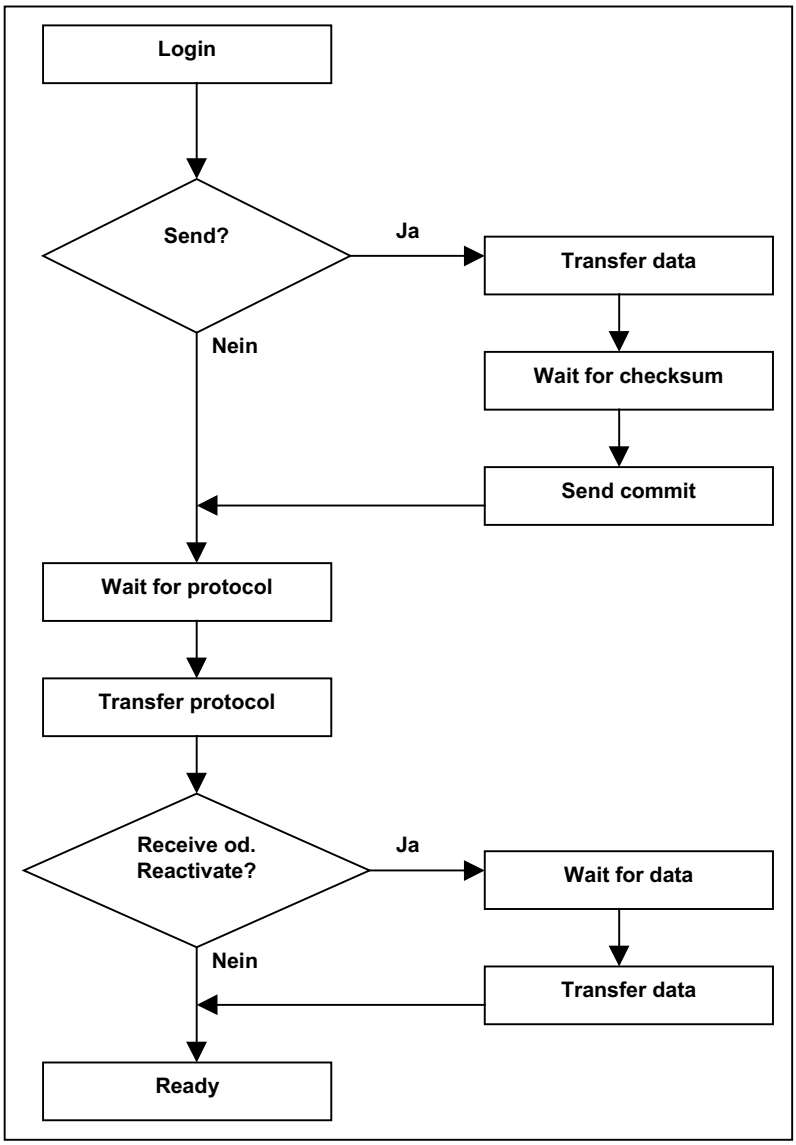

Abbildung 21: Client-Status